NASA Technical Memorandum 88242

Geometry Definition and Grid Generation for a Complete Fighter Aircraft

Thomas A. Edwards

April 1986

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 88242

Geometry Definition and Grid Generation for a Complete Fighter Aircraft

Thomas A. Edwards, Ames Research Center, Moffett Field, California

April 1986

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

GEOMETRY DEFINITION AND GRID GENERATION FOR A COMPLETE FIGHTER AIRCRAFT

Thomas A. Edwards
NASA Ames Research Center
Moffett Field, California 94035
USA

SUMMARY

Recent advances in computing power and numerical solution procedures have enabled computational fluid dynamicists to attempt increasingly difficult problems. In particular, efforts are focusing on computations of complex three-dimensional flow fields about realistic aerodynamic bodies. To perform such computations, a very accurate and detailed description of the surface geometry must be provided, and a three-dimensional grid must be generated in the space around the body. The geometry must be supplied in a format compatible with the grid generation requirements, and must be verified to be free of inconsistencies. This paper presents a procedure for performing the geometry definition of a fighter aircraft that makes use of a commercial computer-aided design/computer-aided manufacturing system. Furthermore, visual representations of the geometry are generated using a computer graphics system for verification of the body definition. Finally, the three-dimensional grids for fighter-like aircraft are generated by means of an efficient new parabolic grid generation method. This method exhibits good control of grid quality, and generates grids about 50 times faster than comparable grids generated via elliptic algorithms.

1. INTRODUCTION

Many applications of computational fluid dynamics (CFD) involve solving the equations governing fluid flow using a finite-difference or finite-volume approximation. The solution procedure commonly involves inverting large, banded matrices in an iterative mode until a steady-state solution is obtained. Depending on the form of the governing equations being solved, the matrices can be scalar tridiagonal (*e.g.*, potential formulations) or block tridiagonal (Euler and Navier-Stokes formulations), where the blocks are commonly 4x4 for two-dimensional problems and 5x5 for three-dimensional problems. To obtain the numerical solution to the flow field, the domain of interest must first be discretized with a grid. Typically, a body-fitted grid is used where the grid boundaries conform to the body surfaces. The grid can be generated by an algebraic interpolation procedure (Ref. 1) or by the solution of partial differential equations (elliptic (Ref. 2,3); parabolic (Ref. 4,5); or hyperbolic (Ref. 6,7)). In any case, some or all of the grid boundaries must be prescribed prior to generating the interior grid points. Thus, the body surface must be defined and a grid of surface points supplied as input to the three-dimensional grid generation procedure.

The numerical solution to the flow field is obtained at each grid node (or at the center of the grid cells for finite-volume formulations), and the accuracy of the solution depends partly on the number and location of the nodes. The Euler and Navier-Stokes equations account for more physical phenomena, but they require a finer computational mesh. Similarly, as the body geometry increases in complexity, more surface grid points are required to maintain an accurate definition of the surface shape. The number of points needed in the computational mesh to solve a given problem, then, is dependent on the level of detail in both the geometry and the flow physics. However, the computer's memory capacity limits the maximum number of points that can be used for a flow calculation. Until recently, computer resources have limited researchers to problems representing a compromise between physical complexity and geometrical complexity: problems involved either complex geometry at the expense of simple flow physics, or complex flow physics at the expense of simple geometry (Fig. 1).

However, the newest and most powerful supercomputers available have for the first time presented the CFD researcher with the tools to attempt problems incorporating both complex flow physics and complex geometry. This, in turn, has created a demand for a new level of accuracy in defining the body geometry. To obtain accurate solutions to viscous flows over complex bodies, the grid must be refined, clustered, and stretched to make the most efficient use of the grid points. When this is done, the distribution of points on the surface of the body must be adjusted accordingly, requiring interpolation of the geometry data base. To minimize errors inherent in the interpolation procedure, the data base should be interpolated as accurately and as few times as possible. The problem of geometry definition includes the modeling of surfaces, along with the process of redistributing points on these surfaces. The resulting surface definition must satisfy the requirements of the three-dimensional grid generator, while maintaining an accurate representation of the actual body surface.

Another requirement in performing geometry definition is error-checking to verify that the geometry is being represented accurately and in a manner consistent with the data base. For most applications, it is sufficient and expedient to verify visually the surface definition by displaying the surface graphically. Toward this end, both shaded-surface and wire-frame displays have proven useful (Ref. 8).

Once the surface definition has prescribed the points on the aircraft surface, a three-dimensional grid must be generated to discretize the space around the aircraft. Grid generation methods frequently involve solving partial differential equations (PDEs) in the coordinate variables, since CFD researchers already have a great deal of experience in the numerical solution of PDEs. Each class of PDE offers certain advantages and disadvantages with regard to the problem of grid generation. Thompson (Ref. 2) has shown that elliptic equations have desirable smoothing properties and permit boundary conditions to be imposed on all boundaries. However, they require iterative solution techniques, which are computationally expensive. Steger and Chausee (Ref. 6) subsequently demonstrated that hyperbolic equations can be used to generate body-fitted grids as well. The solution procedure is noniterative, which is computationally efficient, but boundary discontinuities propagate into the interior field, and outer boundary conditions cannot be prescribed. The present method solves a parabolic equation set, which combines the attributes of the two former methods. The algorithm has a marching (noniterative) solution procedure and smoothing properties, and permits specification of the outer boundary grid. The method was first introduced by Nakamura (Ref. 4), and later was extended by Edwards (Ref. 5).

This paper presents a method for performing geometry definition of fighter-like geometries of interest to current CFD efforts. Throughout the description, the F-16 aircraft will be used to demonstrate the effectiveness of this procedure, since this aircraft is the test case for an ongoing effort to compute transonic, viscous flows over realistic aircraft (Ref. 9). Also, the parabolic grid generation technique will be described in detail. Sample grids have been generated for the F-16 geometry, and the computational efficiency of the parabolic algorithm is compared to that of an elliptic algorithm.

2. GEOMETRY DEFINITION

2.1 CAD/CAM System

In the present study, a computer-aided design/computer-aided manufacturing (CAD/CAM) system is used to define the geometry of a modern fighter jet with minor idealizations. This system uses a VAX 11/780 as a host computer, and consists of an alphanumeric display screen and a graphics display screen (see Fig. 2). The system is suitable for multiple applications --- for example, printed-circuit design, mechanical drawing/drafting, finite-element modeling, and numerically-controlled tool-path computation. Here, the CAD/CAM system performs the surface constructions and geometrical computations of interest to CFD applications. Among the benefits in this approach are that it saves time and effort by using existing, proven technology (Ref. 10) as opposed to developing new software for this specialized application. In addition to being able to perform many geometrical constructions, the system can compute surface intersections, parameterize surfaces, and sample surfaces in several different modes for new point distributions. The system also features a versatile programming language to execute repetitive or computationally laborious tasks.

2.2 Surface Fitting

The input-geometry data base for the example used here consists of (x,y,z) coordinates of points on the cross-sections of various parts of the F-16 aircraft. Figure 3 presents a shaded-surface image of the input geometry. Since the fighter has a very complicated surface, including an inlet, ventral fins, and speed brakes, it was subdivided into 50 simpler components before the cross-sections were taken for the data base. Hence, the input geometry is essentially a file of Cartesian coordinates of cross-sectional points on many small pieces that, when patched together, constitute the exterior surface of the fighter.

The first step in the geometry definition procedure is to fit each subcomponent of the body with B-surfaces. In the syntax of Calma, the term "B-surface" refers to an analytic product surface of one-dimensional B-splines along the parametric directions. This type of surface is somewhat less general than a parametric bicubic (Coons) patch, because it neglects the cross-derivative or "twist" vector at the nodes (Ref. 11). The B-surface is a sufficiently general description for the small, regularly shaped components used in this study. Where appropriate, other surfaces can be generated, such as planar, ruled, Coons, or Bézier surfaces, or surfaces of revolution. Various surface trimming and editing operations are also available, enabling modification of the geometry to suit the constraints of the particular application.

Frequently, it is necessary to make modifications to the actual geometry to reduce the scope of the problem or to focus only on its essential elements. For the F-16 aircraft, initial implementation of the flow solution scheme required that the inlet be faired over and modeled as a streamlined solid surface. This idealization substantially simplifies the grid generation and boundary condition procedures. To do this, the forward section of the inlet was trimmed away using utility functions available on the Calma. A new surface was then defined by generating splines at the boundaries of the region to be faired over. The splines were defined so as to be tangent at both ends to the actual body geometry. Figure 4 shows the new surface definition for the faired-over inlet. Note also that the ventral fins have been removed and that a fillet has been constructed in the nozzle region where the actual geometry has a gap.

2.3 Surface Sampling

Once the input points are fit with B-surfaces, the three-dimensional grid topology must be determined in order to define the type of surface grid needed on the body. The flow solution scheme for the present application uses a cylindrical (O-H) topology, shown in Fig. 5. The fuselage maps into one side of the computational cube, and the wing and horizontal tail correspond to an adjacent side. This means that the surface grid on the fuselage is independent of the grid on the wing and tail surfaces, except that the fuselage cross section points must match up with the wing and tail coordinates where these surfaces intersect. For many transport-type aircraft, a clear demarcation exists between the wing and fuselage, but the fighter in this example has a blended wing-body, which leaves analysts free to define the location of the wing root themselves. This arbitrariness is used to advantage in the present problem since, as seen in Fig. 6, there is a sharp corner in the fuselage cross-section aft of the wing. If the wing root is defined as a straight line parallel to the centerline, the sharp corner in the body cross-section is included as part of the fuselage, introducing the three singular points in the grid shown in Fig. 7a. If, instead, the wing root is defined so as to follow the corner of this shelf, two of the singularities are eliminated, as shown in Fig. 7b. Since topological peculiarities such as this are not easily generalized, they are currently dealt with on a case-by-case basis.

After the analytic surface patches are generated and the topology for the surface grids are determined, the surfaces must be sampled for a distribution of points to supply as a boundary condition for the three-dimensional grid generator. For the current application, axis-normal body cross sections were desired at prescribed body stations. A code was written in DAL, the Calma programming language, which computes the intersections of a plane of constant x with all of the surfaces on the fuselage. A B-spline is constructed at the intersection of the plane and each surface that it intersects. For example, if the fuselage is represented by 30 surfaces, any given plane of constant x may cut, say, 10 of them, so the result of the intersection calculation is 10 B-splines. However, to construct one continuous cross section from these 10 splines, they must be linked together in the proper order and fit as a single spline. This is done by searching from the endpoint of one spline for the nearest spline endpoint (disregarding any splines already linked together) until all 10 splines are ordered end to end. Once the splines are arranged end to end, a single spline is fit through points on the 10 individual splines. These points are obtained by recovering the original control points used to construct the 10 individual splines. An alternative method would be to generate new points by a parametric sampling of the 10 splines, but this would introduce further errors as a result of the additional interpolation. A parametric curve sampling is then calculated on the cross-sectional spline to obtain the fuselage surface grid at that particular body station. The increment of the curve parameter can be adjusted to achieve equally spaced points around the body, or clustering of points near areas of detailed geometry or anticipated large flow gradients. The point coordinates are written into a disk file for subsequent input to the grid generation routine.

Use of the Calma CAD/CAM system in this segment of the geometry definition process has proven useful in representing the complex three-dimensional surfaces associated with flight vehicles. The surface-editing functions are capable of modifying the surface geometry to suit the particular application, and the programming language (DAL) permits nearly the entire process to be automated and made transparent to the user. As stated earlier, the most significant benefit of using the CAD/CAM system for this application is that the technology is immediately available; development time and duplication of effort are circumvented.

3. GEOMETRY VERIFICATION

3.1 Graphics Workstation

Before passing the surface grid to the grid generator and flow solution code, it should be checked for smoothness and continuity. For the present study, two graphical techniques, described below, were employed to verify the geometry. Both exhibit unique advantages and disadvantages with regard to geometry verification. The Silicon Graphics Iris 1500 Workstation was used to display the graphical images. This system features a special hardware transformation system, enabling real-time image manipulation on the display screen, and color display capability. As will be explained below, these tools greatly accelerate the geometry verification step.

3.2 Shaded-surface images

A shaded-surface image is essentially a photograph of the geometrical information used to define the surface. The image is faceted because the location of the surface is defined only at discrete points in space. The shaded-surface display is effective for revealing holes in the surface, and for indicating discontinuities in the surface slope. Degenerate grid cells (cells with fewer than four sides) are difficult to detect using this approach, however.

To produce the shaded-surface image of the surface grid, nonplanar, four-sided polygons are defined on the aircraft surface by connecting the surface grid points. These polygons are then sorted according to the distance from their centroids to a prescribed light source that illuminates them. The orientation of the polygon with respect to the light source determines the shading of the polygon, so that polygons facing the light source are shaded white, and those aligned with the light source are black (Fig. 8). By coloring the polygons in the order of the decreasing distance to the light source, a hidden-surface representation is effected. The shaded polygons are loaded into the

display memory of the Iris workstation, and by virtue of the real-time rotation and translation, a quick visual check of the surface representation is enabled. The shading of the polygons is fixed before they are loaded into display memory since the machine is currently not capable of dynamic reshading. Examples of the display are shown in Figs. 3, 4, and 6.

3.3 Wire-frame images

Another useful display technique is the wire-frame image. In this approach, the points defining the surface shape are connected with straight lines, yielding an image such as that shown in Fig. 9. Since the individual grid cells are outlined on the surface, degenerate cells are easily detected. This type of display is also useful to inspect topological boundaries in the surface grid. A disadvantage of the wire-frame image is that, as the mesh is refined, the image becomes progressively more difficult to visualize. This problem is somewhat alleviated through the use of color, and a significant advancement could be made with hidden-line removal. But the most important element in both display techniques, shaded-surface and wire-frame, is the ability to perform real-time manipulations of the image, and as yet, hidden-line removal cannot be done in real time.

3.4 General remarks

The process of defining the geometry is complete when the surface has been sampled for points in a format compatible with the requirements of the grid generator, and the resulting grid surface has been verified to be free of inconsistencies such as holes, degenerate cells, and inaccurate representation of the geometry. It was mentioned earlier that it is desirable to minimize the number of times the data are interpolated to obtain the final surface grid. By following the foregoing procedure for defining the fighter, the data have been interpolated only twice: once in generating the surfaces from the input data, and a second time in constructing the splines at the surface intersections. Any loss in accuracy using this procedure is compensated for by its generality. Use of the same procedure permits any type of cutting surface to be used to compute surface intersections. This flexibility can be exercised to compute nonaxis-normal sections, and even nonplanar cuts through the body. Moreover, the procedure applies to nearly any continuous surface that can be represented by a set of surface patches. For surfaces with significant twist, the same definition procedure can be applied, making use of the more general Bézier surfaces rather than B-surfaces. A flow chart describing the entire geometry definition and verification process is shown in Fig. 10.

4. GRID GENERATION

The surface grid obtained by the foregoing procedure is acceptable as input to a variety of three-dimensional grid generation methods. Since many existing methods are difficult to use or require a great deal of computing power, analysts are always looking for ways to simplify and expedite the generation of the grid. The method described below implements a new differencing scheme which permits a marching (noniterative) solution procedure. As will be shown, this method results in a substantial savings in both the computer time required to generate the grid and the core memory necessary during the calculation. The algorithm has been used to generate grids about isolated wings for viscous transonic flow computations (Ref. 9) and also for the F-16 geometry described above.

4.1 Equations

The proposed grid generation algorithm is derived from a set of three-dimensional elliptic PDEs. Let (x,y,z) be the coordinates in the physical space and (ξ,η,ζ) be the coordinates in the computational space (Fig. 11). Then the mapping between the two spaces is given by Laplace's equation:

$$\begin{aligned} \xi_{xx} + \xi_{yy} + \xi_{zz} &= 0 \\ \eta_{xx} + \eta_{yy} + \eta_{zz} &= 0 \\ \zeta_{xx} + \zeta_{yy} + \zeta_{zz} &= 0 \end{aligned} \tag{1}$$

These equations are inverted to make (ξ,η,ζ) the independent variables, giving

$$A_1 r_{\xi\xi} + A_2 r_{\eta\eta} + A_3 r_{\zeta\zeta} + 2B_1 r_{\xi\eta} + 2B_2 r_{\eta\zeta} + 2B_3 r_{\zeta\xi} = 0 \tag{2}$$

where $r = x, y,$ or z, and

$$\begin{aligned} A_1 &= (x_\eta^2 + y_\eta^2 + z_\eta^2)(x_\zeta^2 + y_\zeta^2 + z_\zeta^2) - (x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta)^2 \\ B_1 &= (x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta)(x_\zeta x_\xi + y_\zeta y_\xi + z_\zeta z_\xi) - \\ &\quad (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\zeta^2 + y_\zeta^2 + z_\zeta^2) \end{aligned} \tag{3}$$

A_2, A_3, B_2, and B_3 are found by rotating ξ, η, and ζ in A_1 and B_1.

4.2 Differencing Scheme

To form the finite-difference equations from the PDEs, elliptic grid generation algorithms employ central differences to approximate all derivatives:

$$( )_{\xi\xi} \approx \frac{( )_{i+1,j,k} - 2( )_{i,j,k} + ( )_{i-1,j,k}}{(\Delta\xi)^2} \tag{4}$$

The resulting equations are then solved iteratively using relaxation schemes such as successive overrelaxation (SOR) or alternating-direction implicit (ADI) until some convergence criterion is satisfied. In the present study, a "parabolic" differencing scheme is used to approximate the derivatives in η and ζ. In this scheme, the derivative of a quantity at a point corresponding to index n is approximated by a difference between its value at index $(n-1)$ and its value at index $nmax$. These quantities are both known at the first marching step ($n = 2$) from boundary data, permitting a marching solution procedure. For example, the finite-difference approximation to equation (2) above for x at an interior point in the grid with indices i,j,k becomes

$$\begin{aligned}
&A_1 \frac{(x_{i+1,j,k} - x_{i,j,k})/f_i - (x_{i,j,k} - x_{i-1,j,k})/f_{i-1}}{\frac{1}{2}(f_i + f_{i-1})} + \\
&A_2 \frac{(x_{i,jmax,k} - x_{i,j,k})/G_j - (x_{i,j,k} - x_{i,j-1,k})/g_{j-1}}{\frac{1}{2}(G_j + g_{j-1})} + \\
&A_3 \frac{(x_{i,j,kmax} - x_{i,j,k})/H_k - (x_{i,j,k} - x_{i,j,k-1})/h_{k-1}}{\frac{1}{2}(H_k + h_{k-1})} + \\
&2B_1 \frac{x_{i+1,jmax,k} - x_{i-1,jmax,k} - x_{i+1,j-1,k} + x_{i-1,j-1,k}}{(f_i + f_{i-1})(G_j + g_{j-1})} + \\
&2B_2 \frac{x_{i,jmax,kmax} - x_{i,j-1,kmax} - x_{i,jmax,k-1} + x_{i,j-1,k-1}}{(G_j + g_{j-1})(H_k + h_{k-1})} + \\
&2B_3 \frac{x_{i+1,j,kmax} - x_{i-1,j,kmax} - x_{i+1,j,k-1} + x_{i-1,j,k-1}}{(f_i + f_{i-1})(H_k + h_{k-1})} = 0
\end{aligned} \tag{5}$$

where

$$\begin{aligned}
f_i &= \xi_{i+1} - \xi_i & f_{i-1} &= \xi_i - \xi_{i-1} \\
g_{j-1} &= \eta_j - \eta_{j-1} & G_j &= g_j + g_{j+1} + \cdots + g_{jmax-1} \\
h_{k-1} &= \zeta_k - \zeta_{k-1} & H_k &= h_k + h_{k+1} + \cdots + h_{kmax-1}
\end{aligned} \tag{6}$$

The unknown quantities in this equation are the $x_{i-1,j,k}$, $x_{i,j,k}$, and $x_{i+1,j,k}$. All other quantities are known from the boundary grids or previously solved planes. The coefficients A_1, A_2, *etc.*, are lagged one ζ-step to linearize the equations. By approximating derivatives in η and ζ with the parabolic differencing scheme, while still using central differencing for derivatives in ξ, the algorithm can march in both the η- and ζ- directions. A scalar tridiagonal system is obtained for the equations in the ξ-direction. Thus, the formulation is equivalent to the numerical solution of a parabolic PDE with source terms added such that the outer boundary points are obtained exactly when the marching solution reaches the $jmax$ or $kmax$ index.

4.3 Boundary Conditions

Boundary data are required on all six sides of the computational cube. For the purposes of this study, a two-dimensional, parabolic grid generation algorithm generated the outer boundary grids. Body surface grids are supplied from the geometry definition routine described above.

4.4 Clustering

Often, a more accurate flow solution can be obtained by clustering the grid points near regions of anticipated large gradients in the flow variables. To control the spacing between grid lines of the same family, an intermediate computational space is used as described in Ref. 4. The result of clustering grid lines in this space is to make a corresponding displacement of grid lines in the physical space. This is implemented by defining functions f, g, and h that describe the nonuniform values of $(\Delta\xi)$, $(\Delta\eta)$, and $(\Delta\zeta)$, respectively. For a uniformly spaced computational grid, these functions would be unity at every point. To cluster or stretch the grid, the value of the function at a particular point would be the distance from that point to the next point along the particular coordinate direction, as shown in Fig. 12. To generate values for the functions f, g, and h, any convenient stretching formula can be used, such as exponential, cosine, or geometric. These functions are used in forming the finite difference equations so that the differencing schemes become

$$\begin{aligned}
\frac{\partial( )}{\partial\xi} &\approx \frac{( )_{i+1} - ( )_{i-1}}{f_i + f_{i-1}} \\
\frac{\partial( )}{\partial\eta} &\approx \frac{( )_{jmax} - ( )_{j-1}}{G_j + g_{j-1}} \\
\frac{\partial( )}{\partial\zeta} &\approx \frac{( )_{kmax} - ( )_{k-1}}{H_k + h_{k-1}}.
\end{aligned} \tag{7}$$

The computational grid used for the flow calculation is uniformly spaced, and the transformation of variables is contained entirely in the grid metrics.

4.5 Angle Control

Accuracy and stability of the flow calculation are also enhanced by requiring that the grid cells be nearly orthogonal, although for most applications, the only region in which angle control is desired is near the inner boundaries, where the solid body surfaces lie. The angular orientation of grid lines is controlled using a fictitious outer boundary grid to determine the source terms. Since the parabolic algorithm causes the grid lines to march toward the outer boundary grid, the orientation of these grid lines can be controlled by temporarily shifting the outer boundary grid. If the grid is shifted in such a way that the new outer boundary points lie on lines perpendicular to the inner boundary surfaces, the grid lines will march away from the inner surfaces orthogonally, as Fig. 13 shows. In the present study, the shifted outer grid is computed by first calculating the slope of the body surface and the distance from inner boundary points to corresponding outer boundary points. If this distance is designated by $R_{i,j,k}$, then the shifted outer boundary points are given by

$$
\begin{aligned}
x'_{i,j,kmax} &= x_{i,j,1} - \frac{m_x R_{i,j,k}}{\sqrt{1 + m_x^2 + m_y^2}} \\
y'_{i,j,kmax} &= y_{i,j,1} - \frac{m_y R_{i,j,k}}{\sqrt{1 + m_x^2 + m_y^2}} \\
z'_{i,j,kmax} &= z_{i,j,1} + \frac{R_{i,j,k}}{\sqrt{1 + m_x^2 + m_y^2}}
\end{aligned}
\tag{8}
$$

where

$$
m_x = \left.\frac{\partial z}{\partial x}\right|_{surface} \qquad m_y = \left.\frac{\partial z}{\partial y}\right|_{surface} \tag{9}
$$

The shifted outer boundary is scaled so that it lies at the same distance from the inner boundary as the actual outer boundary. This scaling is needed to preserve the desired clustering or stretching of grid lines. As the solution marches toward the outer boundary, the temporary grid is gradually shifted back to the desired outer boundary point distribution. In this manner, angle control is maintained at the body surfaces, and a smooth transition is made to the outer boundary.

5. RESULTS

The parabolic algorithm has been coded to generate O-H grids about wing-fuselage geometries. Using the surface grid generated for the F-16 on the CAD/CAM system, coarse grids have been generated for use in zonal flow solution applications. A grid-embedding approach is used, wherein the coarse outer grid is generated using the parabolic or elliptic scheme and the inner grids are interpolated from the outer grid (Ref. 9). This allows different equation sets to be solved in different grid zones, as appropriate. The Reynolds-averaged, thin-layer Navier-Stokes equations are solved in the zones that have solid surface boundaries, and the Euler equations are solved in the outer zones. Where warranted, it is possible to further refine the grid with additional zones. Since the grid refinement necessary to resolve viscous flows is achieved through grid-embedding, the principal grid is rather coarse and is only slightly clustered near the solid surfaces. Figure 5 shows the O-H topology used. The grid dimensions are 65 (axial) by 20 (circumferential) by 20 (radial). Other detail views are shown in Figs. 14-16. Note also the control of angle and spacing that has been enforced at the body.

6. COMPUTATIONAL EFFICIENCY

An important attribute of the parabolic algorithm is its computational efficiency. Since the solution procedure is noniterative, only one sweep through the domain is required to generate the grid. Also, to generate points on any coordinate plane of constant k, information is needed on the edges of this plane, on the $(k-1)$ plane, and on the $kmax$ plane. Hence, previously computed coordinate planes can be saved permanently, resulting in a substantial economy in core memory requirements. The CPU time used per grid point to generate the grid is roughly independent of the size or refinement of the desired grid; about 62 μsec per grid point on the NASA Ames Cray X-MP22, or 2.44 sec for the 39,000-point grid shown in Figs. 14-16. This is about 50 times faster than a comparable elliptic grid generator that uses SOR on the coarse grids used for these examples. The difference is expected to be even greater as the grid is refined, since the iteration count for elliptic algorithms increases with mesh refinement, while the parabolic algorithm experiences little change in CPU use. Because of the small amount of CPU time required to generate a grid using this scheme, the method is appropriate for use in the interactive development of a grid.

7. CONCLUSIONS

A procedure has been described to define and verify complex aircraft surfaces for use in CFD calculations. By using current technology in CAD/CAM and graphics display systems, the procedure has been made very general and reliable. It has been found that by partitioning the surface into small, regularly-shaped surfaces and using B-surfaces to fit the points, surface sampling can be applied to continuous bodies of arbitrary shape. Also, the geometry verification has been greatly improved by using graphics display systems capable of real-time manipulations of the image. Both the shaded-surface and wire-frame images have proven useful in verifying the accuracy of the surface definition. A parabolic differencing scheme has been implemented to generate three-dimensional grids about fighter-like geometries, and has been shown to control grid spacing and angles in a straightforward manner. The method is also very efficient in terms of computer memory and CPU time.

8. References

1. Smith, Robert E., "Algebraic Grid Generation." *Numerical Grid Generation*, Joe Thompson (ed.), Elsevier Science Publishing Co. , Inc. 1982, pp. 137-170.

2. Thompson, J. , "Elliptic Grid Generation." *Numerical Grid Generation*, Joe Thompson (ed.), Elsevier Science Publishing Co. , Inc. 1982, pp. 79-106.

3. Sorenson, R. L. , "Three-Dimensional Elliptic Grid Generation About Fighter Aircraft for Zonal Finite-Difference Computations." AIAA paper 86-0429, Reno, Jan. 1986.

4. Nakamura, S. , "Marching Grid Generation Using Parabolic Partial Differential Equations." *Numerical Grid Generation*, Joe Thompson (ed.), Elsevier Science Publishing Co. , Inc. 1982, pp. 775-786.

5. Edwards, T. A. , "Noniterative Three-Dimensional Grid Generation Using Parabolic Partial Differential Equations." AIAA paper 85-0485, Reno, Jan. 1985.

6. Steger, J. L. and Chaussee, D. S. , "Generation of Body Fitted Coordinates Using Hyperbolic Partial Differential Equations." *SIAM J. Sci. Stat. Comput.*, Vol. 1, No. 4, Dec. 1980.

7. Steger, J. L. , and Rizk, Y. M. , "Generation of Three-Dimensional Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations." NASA TM-86753, June 1985.

8. Cozzolongo, J. V. , "Aircraft Geometry Verification with Enhanced Computer-Generated Displays." NASA TM-84254, June 1982.

9. Holst, T. L. , Gundy, K. L. , Flores, J. , Chaderjian, N. M. , Kaynak, U. , and Thomas, S. D. , "Numerical Solution of Transonic Wing Flows Using an Euler/Navier-Stokes Zonal Approach." AIAA paper 85-1640, Cincinnati, July 1985.

10. Enomoto, F. , and Keller, P. , "Using a Commercial CAD System for Simultaneous Input to Theoretical Aerodynamic Programs and Wind-Tunnel Model Construction." NASA CP-2272, April 1983.

11. Craidon, C. G. , "A Computer Program for Fitting Smooth Surfaces to an Aircraft Configuration and other Three-Dimensional Geometries." NASA TM X-3206, 1975.

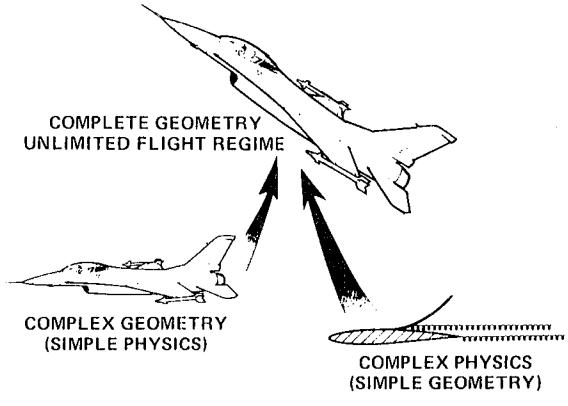

Figure 1. Current computing power enables solution of complex flow physics over detailed, three-dimensional surfaces.

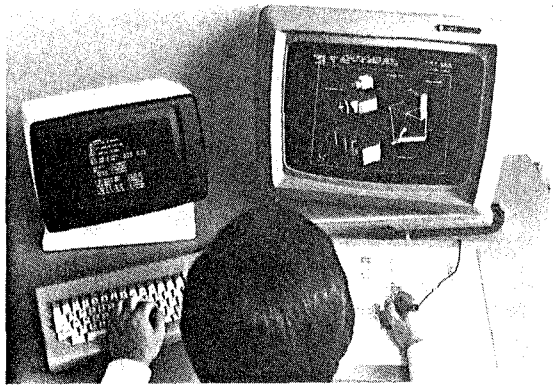

Figure 2. Calma CAD/CAM workstation used for geometrical calculations.

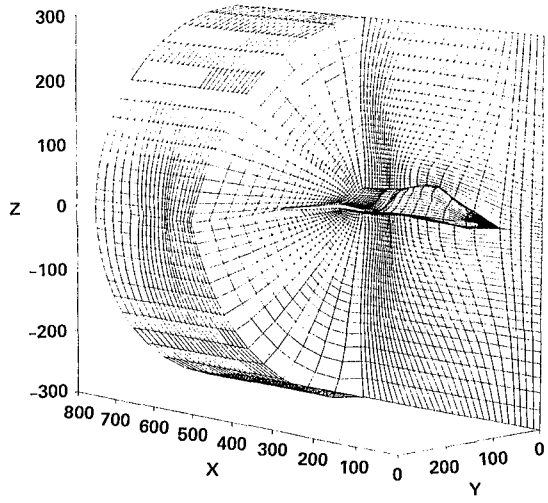

Figure 5. O-H grid topology.

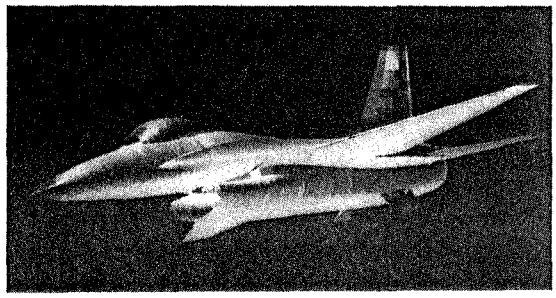

Figure 3. Shaded-surface image of input geometry data base.

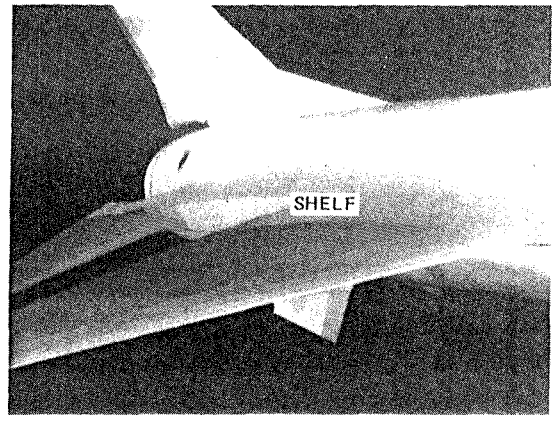

Figure 6. Detail of wing-fuselage junction.

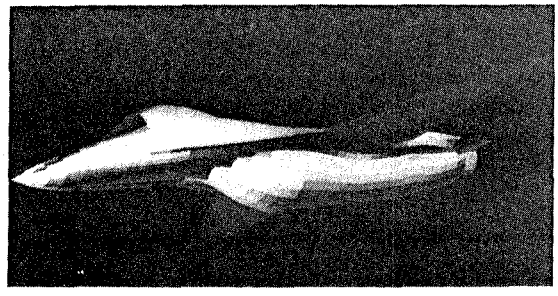

Figure 4. Shaded-surface image of modified geometry with faired-over inlet.

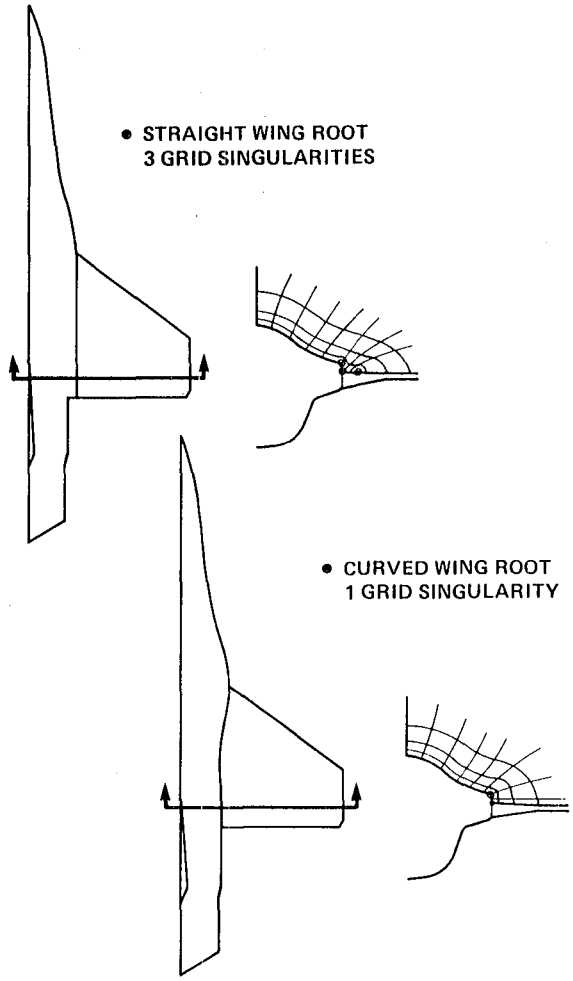

Figure 7. Wing-root definitions and resulting grid singularities.

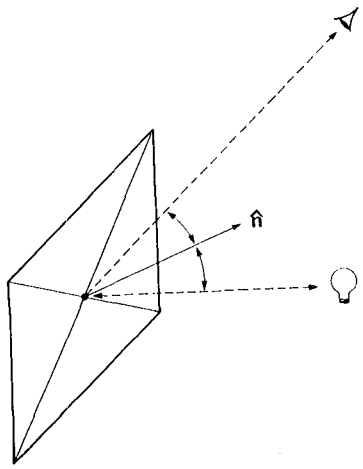

Figure 8. Polygon shading technique.

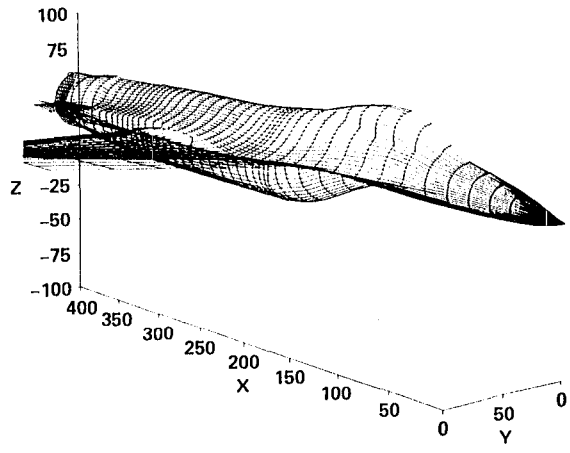

Figure 9. Wire-frame representation of surface grid.

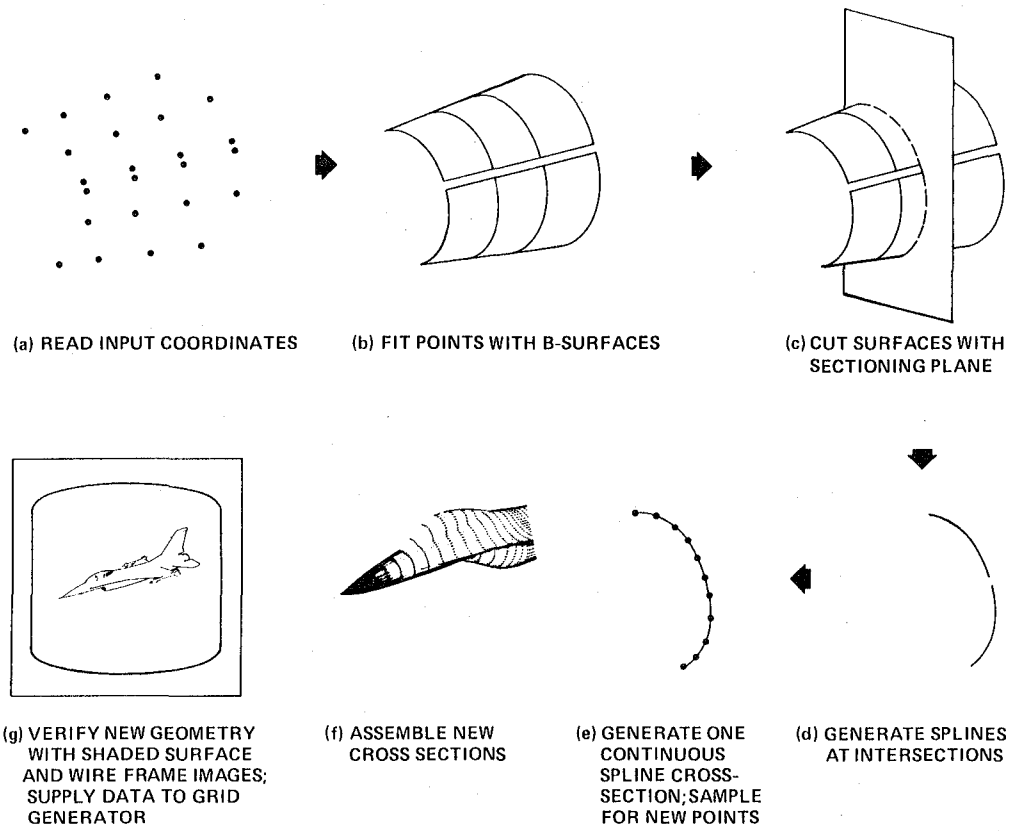

Figure 10. Geometry definition/verification procedure.

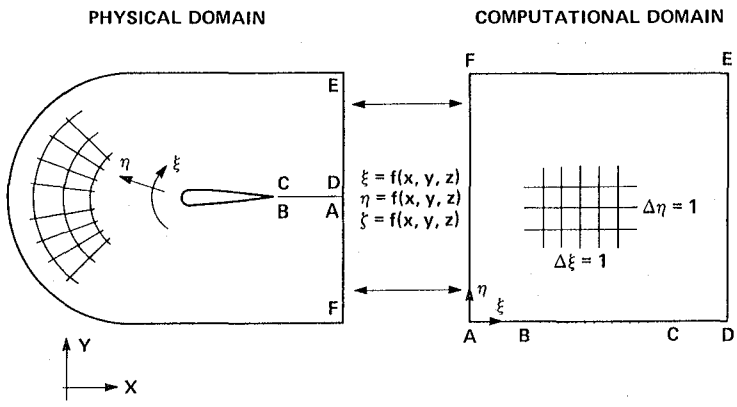

Figure 11. Transformation of variables from physical domain to computational domain.

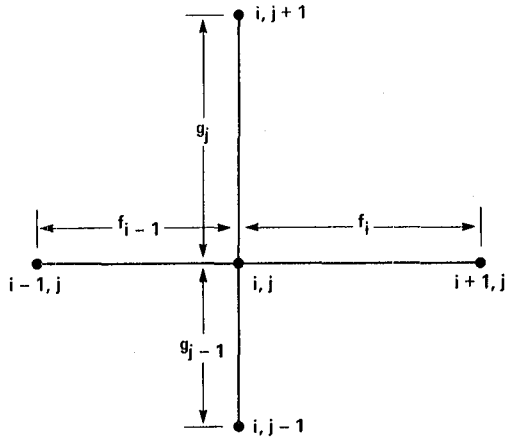

Figure 12. Definition of stretching functions.

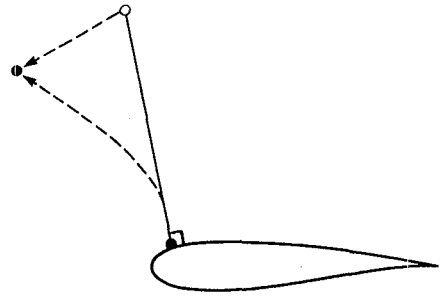

Figure 13. Temporary outer boundary for grid orthogonality.

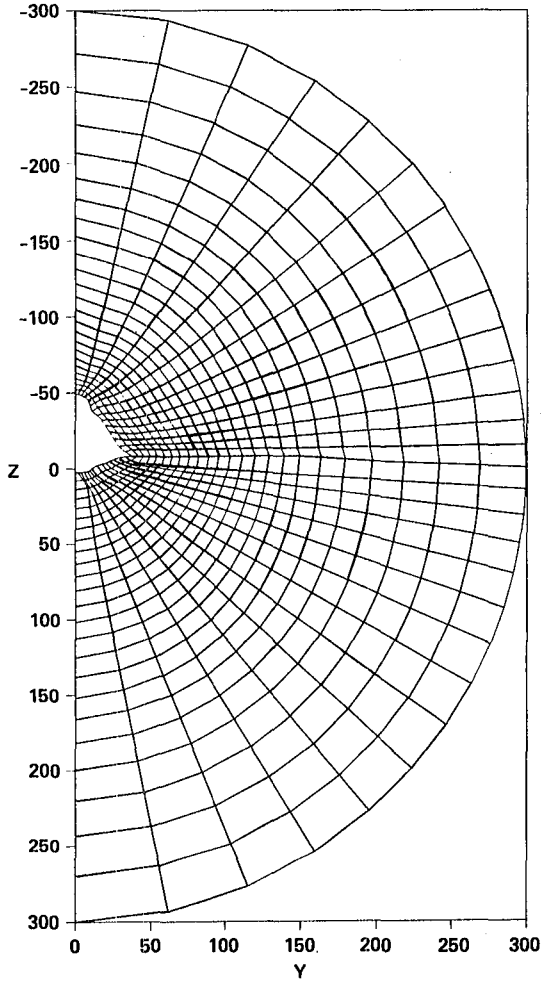

Figure 14. Cross section of grid through canopy and strake.

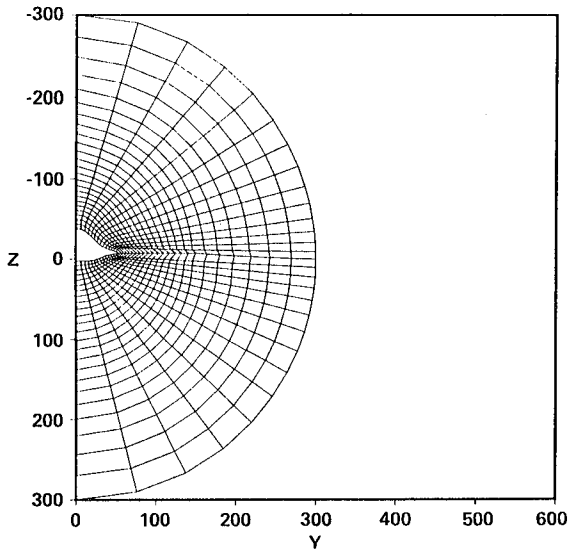

Figure 15. Cross section of grid through wing.

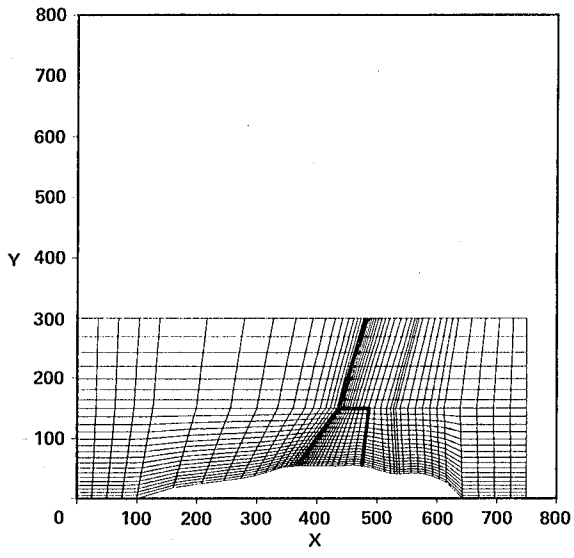

Figure 16. Planform plane of grid with body and wing (top view).

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-88242		

4. Title and Subtitle	5. Report Date
GEOMETRY DEFINITION AND GRID GENERATION FOR A COMPLETE FIGHTER AIRCRAFT	April 1986
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Thomas A. Edwards	A-86208
	10. Work Unit No.
9. Performing Organization Name and Address	
Ames Research Center Moffett Field, CA 94035	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code 505-60

15. Supplementary Notes

Point of Contact: Thomas A. Edwards, Ames Research Center, MS 202A-14, Moffett Field, CA 94035 (415) 694-5369 or FTS 464-5369

16. Abstract

Recent advances in computing power and numerical solution procedures have enabled computational fluid dynamicists to attempt increasingly difficult problems. In particular, efforts are focusing on computations of complex three-dimensional flow fields about realistic aerodynamic bodies. To perform such computations, a very accurate and detailed description of the surface geometry must be provided, and a three-dimensional grid must be generated in the space around the body. The geometry must be supplied in a format compatible with the grid generation requirements, and must be verified to be free of inconsistencies. This paper presents a procedure for performing the geometry definition of a fighter aircraft that makes use of a commercial computer-aided design/computer-aided manufacturing system. Furthermore, visual representations of the geometry are generated using a computer graphics system for verification of the body definition. Finally, the three-dimensional grids for fighter-like aircraft are generated by means of an efficient new parabolic grid generation method. This method exhibits good control of grid quality, and generates grids about 50 times faster than comparable grids generated via elliptic algorithms.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Geometry Definition Grid generation	Unlimited Subject Category - 01

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	14	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document